Sorting algorithms

Data Structures and Algorithms

MEng. Duy Tran Ngoc Bao
Faculty of Computer Science and Engineering
Ho Chi Minh University of Technology, VNU-HCM

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Overview

1. **Sorting concepts**
2. **Selection Sort**
   - Straight Selection Sort
3. **Insertion Sort**
   - Straight Insertion Sort
   - Shell Sort
4. **Exchange Sort**
   - Bubble Sort
5. **Divide-and-Conquer**
   - Quick Sort
   - Merge Sort

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts

Sorting concepts

Selection Sort
- Straight Selection Sort

Insertion Sort
- Straight Insertion Sort
- Shell Sort

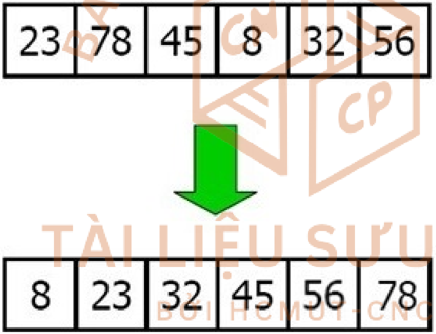
Exchange Sort
- Bubble Sort

Divide-and-Conquer
- Quick Sort
- Merge Sort

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting

One of the most important concepts and common applications in computing.

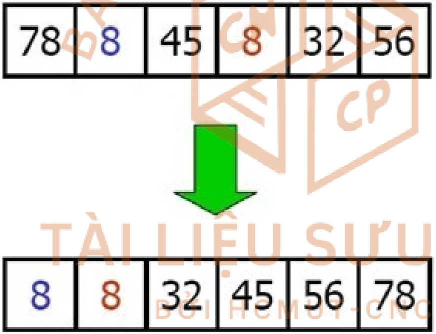


TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Sorting

Sort stability: data with equal keys maintain their relative input order in the output.



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

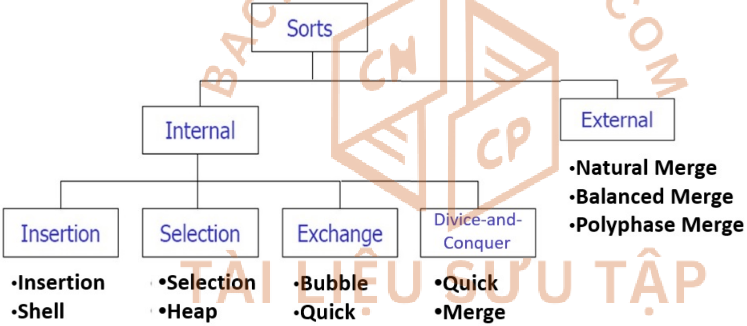
Sorting

Sort efficiency: a measure of the relative efficiency of a sort = number of comparisons + number of moves.

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Sorting



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Selection Sort

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Selection Sort

Idea

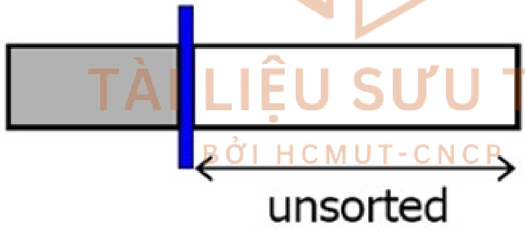
In each pass, the smallest/largest item is selected and placed in a sorted list.

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Straight Selection Sort

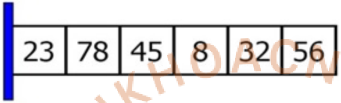
- The list is divided into two parts: sorted and unsorted.
- In each pass, in the unsorted sublist, the smallest element is selected and exchanged with the first element.



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

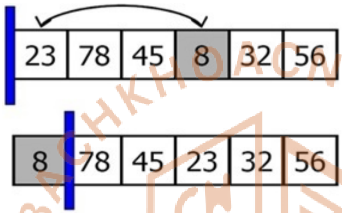
Straight Selection Sort



BACHKHOACNCP.COM
CN CP
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Straight Selection Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting

Duy TNB

Sorting concepts

Selection Sort

Straight Selection Sort

Insertion Sort

Straight Insertion Sort

Shell Sort

Exchange Sort

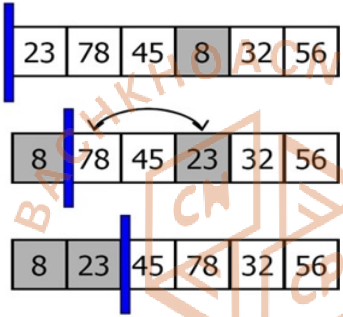
Bubble Sort

Divide-and-Conquer

Quick Sort

Merge Sort

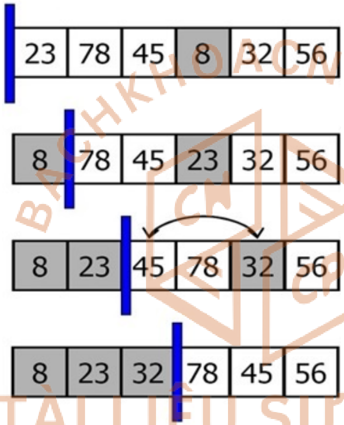
Straight Selection Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

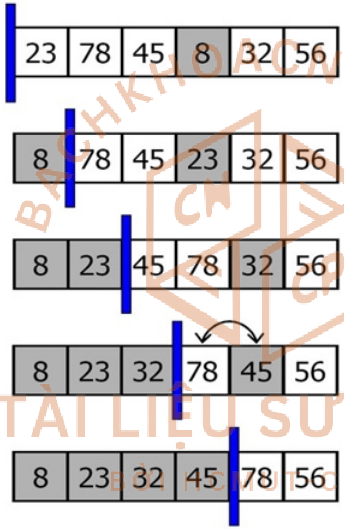
Straight Selection Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

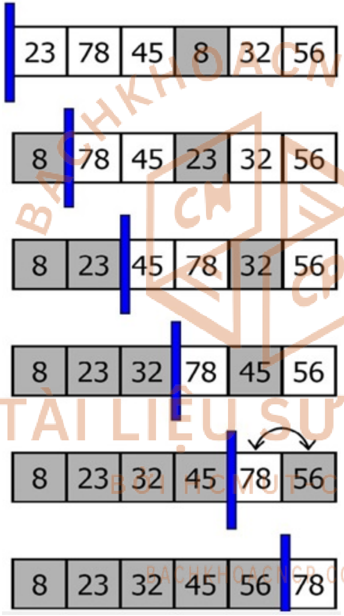
Straight Selection Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Straight Selection Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts

Selection Sort

Straight Selection Sort

Insertion Sort

Straight Insertion Sort

Shell Sort

Exchange Sort

Bubble Sort

Divide-and-Conquer

Quick Sort

Merge Sort

Straight Selection Sort: Pseudocode

```
current = 0
while *current* < *count* - *1* do
    smallest = current
    walker = current + 1
    while *walker* < *count* do
        if *data[walker]* < *data[smallest]* then
            smallest = walker
        end
        walker = walker + 1
    end
    swap(current, smallest)
    current = current + 1
end
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Selection Sort Efficiency

- Straight selection sort: $O(n^2)$

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Sorting concepts

Selection Sort

Straight Selection Sort

Insertion Sort

Straight Insertion Sort

Shell Sort

Exchange Sort

Bubble Sort

Divide-and-Conquer

Quick Sort

Merge Sort

Insertion Sort

BACHKHOACNCP.COM

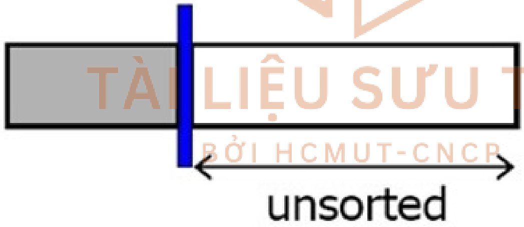
TÀI LIỆU SƯU TẬP

BỞI HCMUT-CNCP

BACHKHOACNCP.COM

Straight Insertion Sort

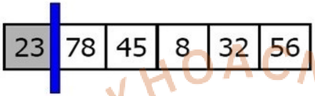
- The list is divided into two parts: sorted and unsorted.
- In each pass, the first element of the unsorted sublist is inserted into the sorted sublist.



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

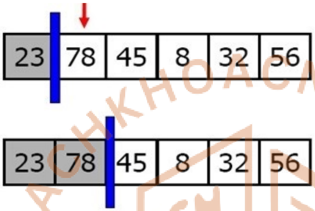
Straight Insertion Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

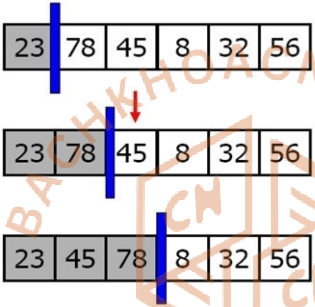
Straight Insertion Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

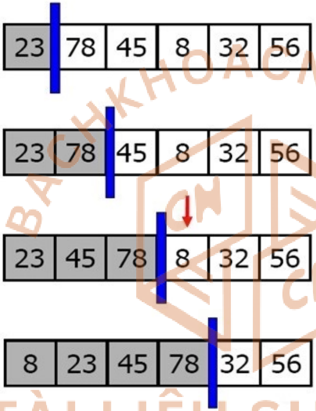
Straight Insertion Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

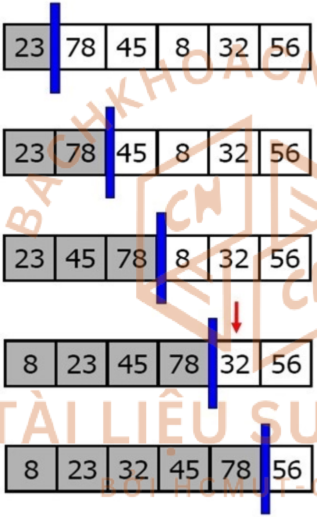
Straight Insertion Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

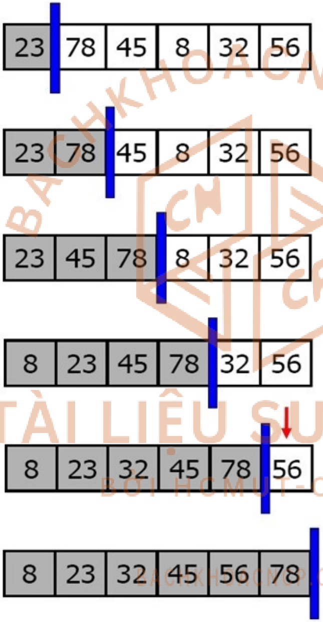
Straight Insertion Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Straight Insertion Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Straight Insertion Sort: Pseudocode

```
if $count > 1$ then
    curr = 1
    while $curr < count$ do
        tmp = data[curr]
        step = curr - 1
        while $step \geq 0$ AND $tmp < data[step]$ do
            data[step + 1] = data[step]
            step = step - 1
        end
        data[step + 1] = tmp
        curr = curr + 1
    end
end
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

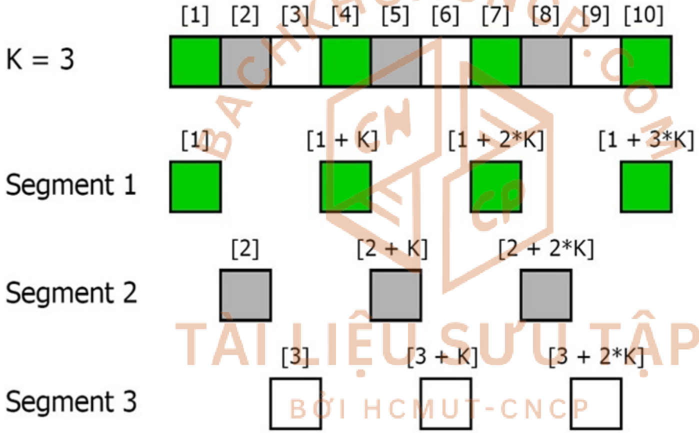
Shell Sort

- Named after its creator Donald L. Shell (1959).
- Given a list of N elements, the list is divided into K segments (K is called the increment).
- Each segment contains $\dfrac{N}{K}$ or more elements.
- Segments are dispersed throughout the list.
- Also is called diminishing-increment sort.

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

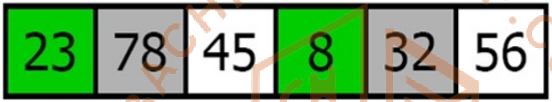
Shell Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Shell Sort



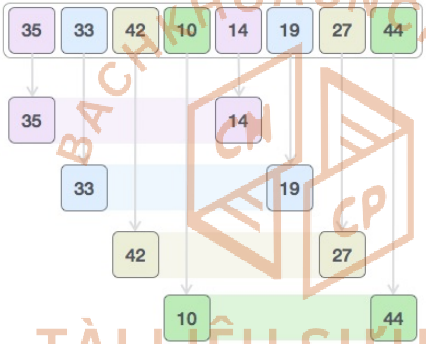
- For the value of K in each iteration, sort the K segments.
- After each iteration, K is reduced until it is 1 in the final iteration.

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

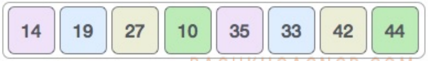
Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Example of Shell Sort

$K = 4:$



Result:

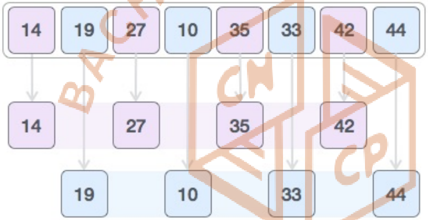


BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

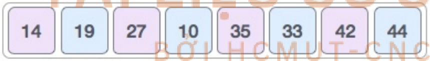
Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Example of Shell Sort

$K = 2:$



Result:

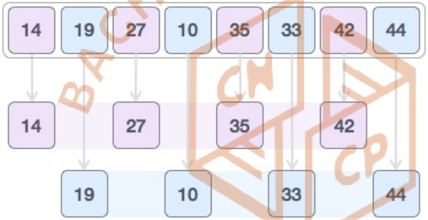


BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

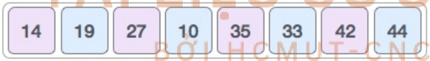
Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Example of Shell Sort

$K = 2:$



Result:

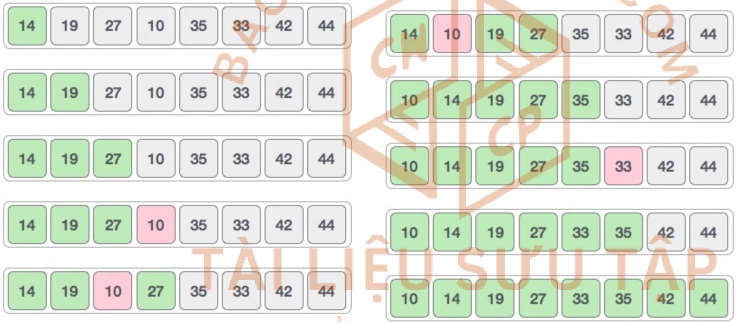


BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Example of Shell Sort

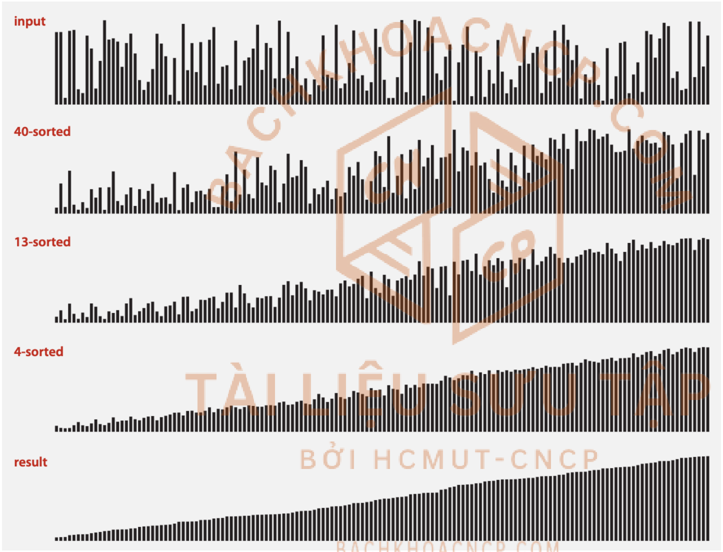
$K = 1:$



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Shell sort: Visual trace



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts

Selection Sort
Straight Selection Sort

Insertion Sort
Straight Insertion Sort
Shell Sort

Exchange Sort
Bubble Sort

Divide-and-Conquer
Quick Sort
Merge Sort

Choosing incremental values

- From more of the comparisons, it is better when we can receive more new information.
- Incremental values should not be multiples of each other, other wise, the same keys compared on one pass would be compared again at the next.
- The final incremental value must be 1.

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Choosing incremental values

- Incremental values may be:
  $1, 4, 13, 40, 121, \ldots$
  $k_t = 1$
  $k_{i-1} = 3 * k_i + 1$
  $t = \lfloor \log_3 n \rfloor - 1$

- or:
  $1, 3, 7, 15, 31, \ldots$
  $k_t = 1$
  $k_{i-1} = 2 * k_i + 1$
  $t = \lfloor \log_2 n \rfloor - 1$

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Shell Sort: Pseudocode

```
k = next_increment()
while $k \geq 1$ do
    segment = 1
    while $segment \leq k$ do
        sort_segment(segment, k)
        segment = segment + 1
    end
    k = next_increment()
end
```

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Shell Sort - Sort Segment: Pseudocode

sort_segment(val segment <int>, val k <int>)

```
curr = segment + k
while curr < count do
    temp = data[curr]
    step = curr - k
    while step >=0 AND tmp < data[step] do
        data[step + k] = data[step]
        step = step - k
    end
    data[step + k] = tmp
    curr = curr + k
end
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Insertion Sort Efficiency

- Straight insertion sort:
  $$f(n) = \frac{n(n+1)}{2} = O(n^2)$$
- Shell sort:
  $O(n^{1.25})$ (Empirical study)

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

BACHKHOACNCP.COM

Exchange Sort

TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Exchange Sort

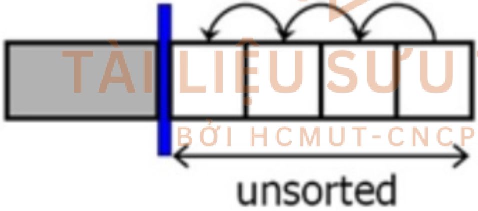
- In each pass, elements that are out of order are exchanged, until the entire list is sorted.
- Exchange is extensively used.

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Bubble Sort

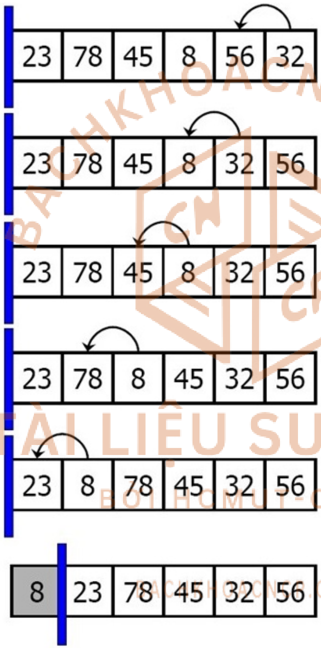
- The list is divided into two parts: sorted and unsorted.
- In each pass, the smallest element is bubbled from the unsorted sublist and moved to the sorted sublist.



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

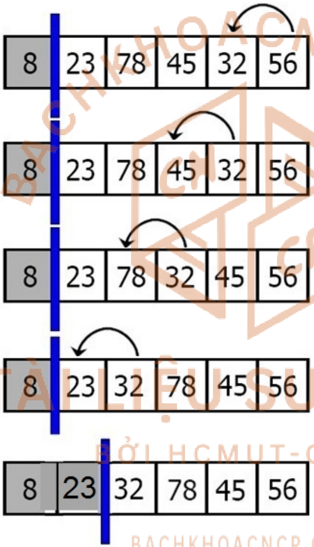
Bubble Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Bubble Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Bubble Sort

```
curr = 0
flag = False
while $curr < count$ AND $flag = False$ do
    step = count - 1
    flag = True
    while $step > curr$ do
        if $data[step] < data[step - 1]$ then
            flag = False
            swap(data[step], data[step - 1])
        end
        step = step - 1
    end
    curr = curr + 1
end
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Exchange Sort Efficiency

- Bubble sort:

$$f(n) = \frac{n(n+1)}{2} = O(n^2)$$

BACHKHOACNCP.COM
CN
CP
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Divide-and-Conquer

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Divide-and-Conquer Sort

```
Algorithm DivideAndConquer()
if the list has length > 1 then
    partition the list into lowlist and
      highlist
    lowlist.DivideAndConquer()
    highlist.DivideAndConquer()
    combine(lowlist, highlist)
end
End DivideAndConquer
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Divide-and-Conquer Sort

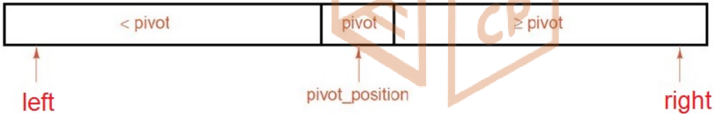
	Partition	**Combine**
Merge Sort	easy	hard
Quick Sort	hard	easy

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Quick Sort

Given a pivot value, the partition rearranges the entries in the list as the following figure:



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Quick Sort

```
1 Algorithm QuickSort()
2 Sorts the contiguous list using quick sort.

3 recursiveQuickSort(0, count - 1)
4 End QuickSort
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Quick Sort

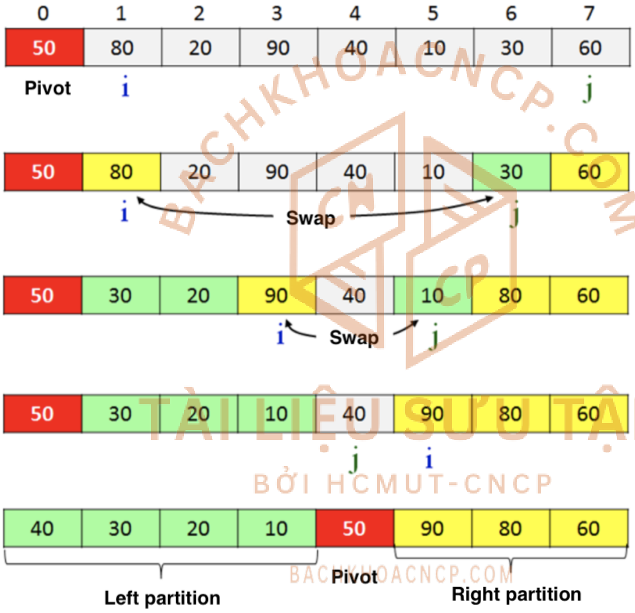
```
1 Algorithm recursiveQuickSort(val left <int>, val right <int>)
2 Sorts the contiguous list using quick sort.
3 Pre: left and right are valid positions in the list
4 Post: list sorted

5 if $left < right$ then
6     pivot_position = Partition(left, right)
7     recursiveQuickSort(left, pivot_position - 1)
8     recursiveQuickSort(pivot_position + 1, right)
9 end
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Partitioning



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

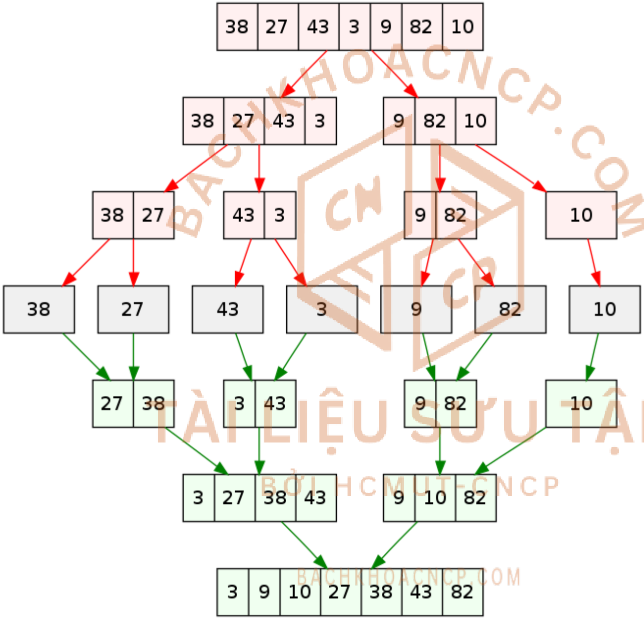
Quick Sort Efficiency

- Quick sort: $O(n \log_2 n)$

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Merge Sort



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts

Selection Sort
Straight Selection Sort

Insertion Sort
Straight Insertion Sort
Shell Sort

Exchange Sort
Bubble Sort

Divide-and-Conquer
Quick Sort
Merge Sort

Merge Sort

```
Algorithm MergeSort()
Sorts the contiguous list using merge sort.

recursiveMergeSort(arr, 0, arr.length() - 1)
End MergeSort
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

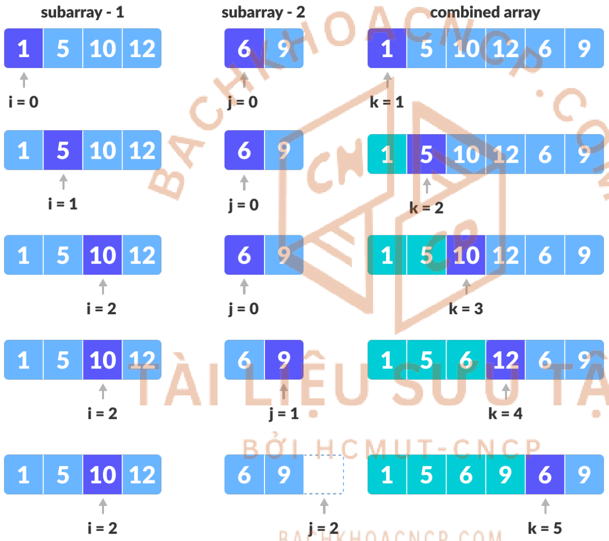
Merge Sort

```
1 Algorithm recursiveMergeSort(ref arr
  <array>, val hi <int> , val lo <int>)
2 if $hi > lo$ then
3     mid = lo + (hi - lo) / 2
4     recursiveMergeSort(arr, lo, mid)
5     recursiveMergeSort(arr, mid + 1, hi)
6     merge(arr, lo, mid, hi)
7 end
8 End recursiveMergeSort
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Merge - Combine operation



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Merge

```
Algorithm merge(ref arr <array>, val hi <int>, val mid <int>, val lo <int>)

Allocate new array aux with the length of arr.
k = lo
while $k \geq hi$ do
    aux[k] = arr[k] i++
end
k = lo
while $k \geq hi$ do
    if $i > mid$ then
        a[k] = aux[j++]
    end
```

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Sorting concepts
Selection Sort
Straight Selection Sort
Insertion Sort
Straight Insertion Sort
Shell Sort
Exchange Sort
Bubble Sort
Divide-and-Conquer
Quick Sort
Merge Sort

Sorting concepts

Selection Sort
Straight Selection Sort

Insertion Sort
Straight Insertion Sort
Shell Sort

Exchange Sort
Bubble Sort

Divide-and-Conquer
Quick Sort
Merge Sort

THANK YOU.

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM